

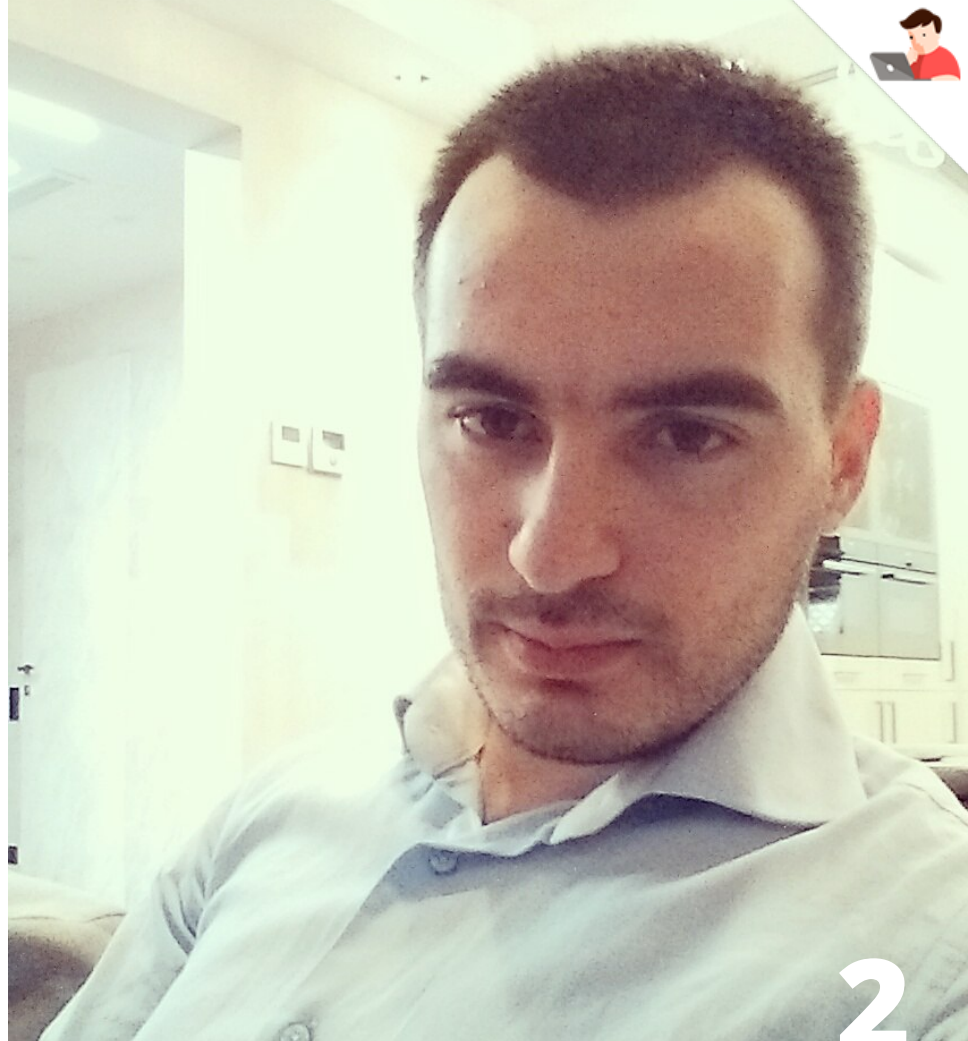
Объекты и массивы

Ведущий вебинара

Сергей Мелюков

Круг интересов: Backend, Frontend, GameDev, MobileDev

Место работы: Frontend разработчик профессиональных инструментов Avito



Содержание



1. Что такое объект
2. Что такое массив
3. Функции для работы с массивами
4. Методы Object
5. Динамическое создание свойств у объектов

The background features a stylized illustration of mountains and a cloud. The mountains are composed of several overlapping triangles in shades of light gray and white, creating a sense of depth. A single, soft white cloud is positioned on the left side, partially overlapping the mountain peaks. The entire scene is set against a light gray background. In the top right corner, there is a yellow L-shaped line. In the bottom left corner, there is another yellow L-shaped line. A solid yellow horizontal bar runs along the bottom edge of the slide.

Объекты

Что такое объект?

Объект - это один из типов данных в JS.

Объект можно представить себе как *набор свойств*

Каждое свойство объекта имеет **имя** и **значение**

Любой тип данных в JS *пытается* *подражать* объекту своим поведением, следовательно можно сказать, что **всё в JS есть объект**





Работа со свойствами объекта

Свойства объекта можно **перечислить** при помощи оператор **for...in**

Чтение свойств осуществляется через **точку** или **квадратные скобки**

Чтение через **квадратные скобки** чаще всего применяется, когда имя свойства *заранее не известно*

Удалить свойство можно при помощи оператора **delete**

Проверить свойство на существование можно, *сравнив* значение свойства с **undefined**, или используя оператор **in**

Методы Object



Object.hasOwnProperty (объект, имя свойства) имеет ли объект указанное свойство

Object.keys (объект)	список имен свойств
-----------------------------	---------------------

Object.toString ()	функция объекта, которая будет выполняться каждый раз, при попытке представить объект как строковое значение
---------------------------	--

Object.valueOf ()	функция объекта, которая будет выполняться каждый раз, при попытке представить объект как не строковое значение
--------------------------	---



Динамическое создание свойств

Свойства у объектов могут быть созданы динамически при помощи метода **Object.defineProperty**(**объект**, **имя свойства**, **параметры**)

Параметры могут быть следующие:

configurable	можно ли будет изменить параметры свойства
enumerable	будет ли свойство доступно для оператора for
value	значение по умолчанию
writable	можно ли перезаписывать значение у свойства
get	функция, выполняющаяся при попытке получить значение свойства
set	функция, выполняющаяся при попытке установить значение свойства

The background features a stylized mountain range with several peaks in shades of gray. A white cloud is positioned on the left side, partially overlapping the mountains. In the top right corner, there is a yellow L-shaped line. In the bottom left corner, there is a yellow L-shaped line. A solid yellow horizontal bar runs along the bottom edge of the slide.

Массивы



Массивы

Массив это - **нумерованный** набор значений

Нумерация начинается **с нуля**

Длину массива можно узнать, обратившись к свойству **length**

Перебрать все элементы массива можно при помощи оператора **for**

Доступ к элементам массива осуществляется при помощи **квадратных скобок**

Доступ к элементам массива осуществляется при помощи двух **квадратных скобок**, следующих друг за другом

Удаление элементов массива *рекомендуется* осуществлять методом **splice**

Методы Array



concat()

объединение массивов

every()

удовлетворяют ли все элементы массива заданному условию

filter()

создаёт новый массив со всеми элементами, прошедшими проверку

indexOf()

возвращает первый индекс, по которому данный элемент может быть найден в массиве или -1, если такого индекса нет

forEach()

выполняет указанную функцию для каждого элемента в массиве



Методы Array

<code>join()</code>	объединяет все элементы массива в строку
<code>lastIndexOf()</code>	возвращает <i>последний</i> индекс, по которому данный элемент может быть найден в массиве или -1, если такого индекса нет
<code>map()</code>	создаёт <i>новый</i> массив с результатом вызова указанной функции для каждого элемента массива
<code>pop()</code>	удаляет последний элемент из массива и возвращает его значение
<code>push()</code>	добавляет один или более элементов в конец массива и возвращает новую длину массива



Методы Array

reduce()

применяет функцию к каждому значению массива (слева-направо), сводя его к одному значению

reduceRight()

применяет функцию к каждому значению массива (справа-налево), сводя его к одному значению

reverse()

переворачивает массив (изменяет массив)

shift()

удаляет первый элемент из массива и возвращает его значение

slice()

возвращает копию указанной части массива

Методы Array



some()

удовлетворяет ли хотя бы один элемент массива заданному условию

splice()

изменяет содержимое массива, добавляя/удаляя элементы

unshift()

добавляет один или более элементов в начало массива и возвращает новую длину массива

Еще немного о массивах

Строки частично ведут себя как массивы:

- у строки есть свойство **length**
- к символам внутри строки можно обращаться при помощи **квадратных скобок**

Объекты тоже могут вести себя как строки:

- у таких объектов присутствует свойство **length**
- именами свойств таких объектов являются **цифры**

Примером таких объектов, является объект **arguments**, который доступен внутри функций и содержит в себе переданные в функцию аргументы.



Время ваших вопросов